

REMARKS

Applicants and their attorney thank the examiner for his continuing assistance in the prosecution of the present application.

Claims 2-37, 39-40, 42, 44-48 and 50-52 were examined. No claims were cancelled or are newly presented.

Drawings:

The examiner requires clean copies of the drawings be filed. Applicant submits clean copies of previously amended drawings.

Claims amendments:

Claims 50, 51 and 52. have been amended. Support for the amendments is found as follows:

Paragraph 0038 discloses the use of semantic analysis to decompose test cases into application state, external interaction sequences and input data. Paragraph 0039 discloses application objects, attribute definitions, definitions of methods and events and definitions of effects of events. Paragraph 0043 discloses abstraction representation of test cases, and the application metadata repository. Paragraph 0045 discloses the computer system, processor coupled to the memory that stores program instruction for converting test cases to an abstraction representation that includes application state, external interaction sequences and input data. Paragraph 0045 also discloses that the database stores abstraction representation of test cases. Paragraph 0049 discloses application states, external interaction sequences, input data and rules for data driven test case generation. Paragraph 0053 discloses platform specific mapping. Paragraph 0049 also discloses generation of test scripts for multiple test execution environments and that mappings are provided to support additional platform independence. Paragraph 0055 discloses logic provides the components of a test case definition, namely, application states, external interaction sequences and input data. Paragraph 0056 discloses that the logic provides rules for selection of components of test case definition, namely application states, external interaction sequences, input data, and rules for data driven test case generation.

Figure 3 illustrates external interactions and application state, application object and attribute. Figure 6 illustrates, application object model, application object type, application object attribute, application object event, application event parameter and application event effect. Figures 1 and 9 illustrates the relation of the abstract format and the memory. The application state, interaction sequences and input data are in the memory. Figure 10 illustrates the test case repository, external data sources, the application metadata repository, and the abstract form that includes state, control and data.

In one embodiment of the present invention, as set forth in claim 50, a method is provided for transforming test cases that are converted from a source test script to an abstract representation. At least one test case is imported that is written in one or more scripting language. The test case is converted to an abstract representation using semantic analysis without changing or deleting an original test case based on an application object model. The abstract representation has at least three separate components including, an application state, external interaction sequences and input data, The application object model is a metadata representation for a modeling application under test and includes components selected from application object type definitions for application objects, attribute definitions for each application object type, definitions of methods and events that are supported by each application object type and definitions of effects of events on an application state. The abstract representation is stored in a database. At least one platform-specific test script is generated that is based on the abstract representation stored in the database using a platform-specific mapping. The platform-specific mapping includes a language mapping and an environmental mapping.

Winrunner fails to, (i) transform test cases to an abstract representation using semantic analysis, (ii) provide an abstract representation has at least three separate components including, an application state, external interaction sequences and input and (iii) generate at least one platform-specific test script based on the abstract representation stored in a database using a platform-specific mapping.

Winrunner creates tool dependency and reduces the ability to share tests.
WinRunner does not use semantic analysis to covert test cases to abstract representation.

CONCLUSION

It is submitted that the present application is in form for allowance, and such action is respectfully requested. The Commissioner is authorized to charge any additional fees which may be required, including petition fees and extension of time fees, to Deposit Account No. 50-4634 (Docket No. SYM-0002).

Respectfully submitted,
GOODWIN | PROCTER LLP

Date: 2.25.09

135 Commonwealth Drive
Menlo Park, CA 94025
Tel: 650/752-3106
Customer No. 77845


Paul Davis, Reg. No. 29,294